Proceedings of the Ninth International Space Syntax Symposium
Edited by Y O Kim, H T Park and K W Seo, Seoul: Sejong University, 2013

NETWORK COMMUNITIES IN THE VISIBILITY GRAPH: A new method for the discretization of space

045

Yongha Hwang
University of Michigan
e-mail: yonghah@umich.edu

Abstract

Although spatial decomposition is a commonly required process in the analysis of space, only a few rigorous methods such as convex partitioning have been proposed and used in practice. This paper discusses the shortcomings of current methods such as operational arbitrariness during the decomposing process and inherent difficulties in automation. Then, this paper proposes a new method for discretizing space based on maximizing the modularity of the visibility graph of a given floorplan. It is shown that the new method is robust to operational arbitrariness and can be easily automated. Moreover, the method decomposes space into subspaces based on the global property of the spatial network in contrast to the traditional methods using only local properties. Another advantage of the new method is the capability of adjusting the resolution of decomposition according to a researcher's purpose of analysis. We expect that the new method is useful to construct a spatial network as well as to analyze the deep structure of given space.

Keywords: *discretization of space, community detection, space decomposition, building layout, convex map*
Theme: *Modeling and Methodological development*

1. Introduction

Decomposing space into subspaces and translating it into a discrete system gives benefits to researchers. Firstly, representing space as a discrete system helps researchers to easily understand the spatial structure by decreasing its complexity. Secondly, once the system becomes discrete, analytic methods that are only applicable to discrete systems - for example, network analysis - become available to apply. The benefit of describing space as a discrete system in terms of analytic power has been well shown by Space Syntax over a few decades. However, to translate a spatial layout into a discrete system is not an easy task (Peponis and Wineman 2002). Only a few rigorous methods for the discretization of space have been developed so far such as convex partitioning (Hillier and Hanson 1984) and e-partitioning (Peponis et al. 1997), both of which have limitations.

Firstly, current methods requires a certain degree of approximation in spatial structure or minor violations of protocols when the methods are applied to space with, for example, a curved wall, a concave wall, a free-standing column, and a small indent. This looks trivial when such elements are ignorable. However, this becomes a source of arbitrariness in decomposition when those elements are no longer ignorable.

Secondly, current methods consider only how well a ‘cut’ of space achieves the internal completeness of resultant subspace such as convexity. The methods do not consider how well the cut *separates* adjacent subspaces. For example, what convex partitioning cares is whether resultant subspaces by the cut are convex, not how effectively the cut separates the two subspaces. In addition, current methods does not utilize the global property of spatial structure during the partitioning process; instead, only local spatial structure is used in determining the location of cut.

Lastly, current methods does not have a well-defined way to adjust the ‘resolution’ of decomposition. A researcher working on an analysis of a huge building may want to ignore trivial violations of the convexity rule in convex partitioning to reduce unnecessary bias in the building’s spatial configuration. Or the researcher may want to merge small e-spaces with tiny differences in visual information to reduce unnecessary complexity beyond one’s purpose of an analysis. However, there is no ‘native’ method or well-established process for current methods to adjust the coarseness or fineness of decomposition. What a researcher can usually do is to ignore trivial violations or to merge small subspaces on the fly at the risk of arbitrariness.

The aim of this study is to propose a new method for space discretization that is readily applicable to the real world’s space with curved walls, columns, and indents; that provides a partitioning process considering the quality of spatial separation with a view to global spatial structure; and that enables to adjust the level of analytic resolution within its process. The core idea of this method is to apply a community detection algorithm to the visibility graph of a floor plan so that a ‘community’ of grid points in the visibility graph becomes each partitioned space.

2. Background

2.1 Need of area –based decomposition

There has been three ways to translate a floorplan into a discrete system: line-based decomposition, area-based decomposition, and point-based decomposition. Each approach has its own strengths and weaknesses.

Line-based decomposition usually means axial line decomposition. Axial line representation of space is suitable for linear spaces such as streets or roads and makes the calculation of the number of turns easier. For indoor spaces, it is fairly useful for closed plan with well-defined corridor spaces and cell-type offices. However, axial line decomposition is often too coarse for indoor spaces, and it does not appropriately represent the spatial structure of open plan.

Since Visibility Graph Analysis (VGA) technique was introduced (Turner and Penn 1999), area-based analysis such as convex map analysis is sometimes looked as an inferior alternative to point-based analysis such as VGA. VGA has two strengths over area-based analysis. Firstly, it has much higher 'resolution' of analysis and hence suitable for very detailed analysis. Secondly, it requires much less effort to build a spatial network than convex partitioning. Unlike convex partitioning that requires to manually draw convex polygons and link them in order to build a spatial network, VGA has an automated process of building a spatial network.

Visibility graph analysis, however, tends to over-estimate a large room's spatial properties under some circumstances. This mostly happens when a researcher takes the average property of points in the space as the property of the space, which is very common practice. Let's think about an imaginary floor with two rooms connected by a corridor space (see Figure 1). The floor has a symmetrical layout except for the size of the two rooms. We are interested in mean depth of each room. Let's build a visibility graph on the grid points in the floor. From a grid point in a room, all of the grid points in the same room are one step's away; the grid points in the corridor are two steps' away; finally, the grid points in the other room are three steps' away, which is the maximum distance. Then, the average of mean depth of the grid points in the large room would be significantly smaller than that of the grid points in the small room because a grid point in the large room has more grid points at one step's away and less grid points at two steps' away than a grid point in the small room. In Figure 1, for example, from point A in the large room, 29 points are at one step's away, 10 points at two steps' away, and 10 points at three steps' away. So the mean depth of point A is 1.61. All points in the large room will have this value if we ignore trivial exceptions on the border. From Point B in the small room, 10 points are at one step's away, 10 points at two steps' away, and 29 points at three steps away. So the mean depth of point B is 2.39 and all points in the small room will have this value. Thus, if we take the average of mean depth of the grid points in each room as the representative value of mean depth, or accessibility, of each room, then the large room becomes much more accessible than the small room even though the two rooms occupy symmetrical locations.[1] Such bias happens solely because the large room is large and small room is small, not because the two rooms occupy such spatial locations. And the bias would not happen if we decomposed the floor into three 'areas', instead of 49 'points'.

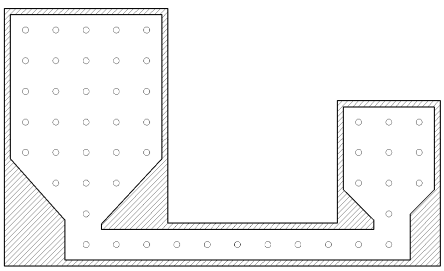


Figure 1: Imaginary floor with two rooms

[1] The bias would not go away when we calculate mean depth without the points in the same room. The bias persists because the size of the other room matters. For example, if we calculate the mean depth of a point in the large room excluding the points in the same room, the mean depth is (10*1 + 10*2)/20 = 1.5. And the mean depth of a point in the small room is (10*1 + 29*2)/39 = 1.74. Still, the large room has smaller mean depth.

Area-based decomposition is also useful when we need to count spatial events. For example, we often need to answer a question like “which space is the most frequently used for X?” or “where did X happen the most frequently?” because of very practical reasons. Also, by describing spatial events as frequency, a researcher can use statistical techniques for count variables. In order to count such frequencies, we need a ‘bin’ to put events in. Area-based decomposition can provide appropriate bins for such purpose.

2.2 Traditional ways of area-based decomposition

Convex partitioning

Currently, the most widely used rigorous methods for the decomposition of indoor space is convex partitioning. This method aims to decompose a floor into the fewest number of convex polygons required to cover the floor (Hillier and Hanson 1984). Partitioning space into convex polygons is a very powerful concept because the convexity of space guarantees that any two people in the space can see each other. However, the process of convex decomposition is difficult to automate because it is known as NP-complete problem[2] (Karp 1972). Thus convex partitioning has been mostly conducted manually.

Manual convex decomposition is vulnerable to the arbitrariness of an operator who decomposes a floor because it is almost impossible to strictly apply the convexity rule under some circumstances.

Firstly, convex decomposition is not applicable at all to space with a concave curved wall like Floor A in Figure 2. To decompose such a floor to convex spaces, we have to approximate the concave curved walls with a series of straight walls. Thus the composition of space is mainly determined by the way how the curved walls are approximated with straight walls. This means considerable amount of arbitrariness might be involved in the decomposition process.

Secondly, convex decomposition does not handle well a space with small ‘indents’ or ‘bulges’. For example, the upper room on Floor C in Figure 2 would have only one convex space if there were no indents. However, because of the indents, the room should be split into three convex spaces lest convexity rule should be violated. If the indents become smaller, we would be more tempted to ignore such indents and to violate the convexity rule. This is another source of arbitrariness.

Thirdly, more complicated case happens when we try to decompose space with a column. Let’s assume that we are going to decompose a room with free standing columns like Floor B in Figure 2 using convex partitioning method. There are three options. The first option is to literally follow convexity rule and to get 12 convex spaces[3] from the simple room. The second option is to simply ignore all columns and to get only one convex space as if the columns are nothing to do with the separation of space. The last option is to ignore the columns’ size without forgetting the existence of the columns. This option gives us two or three convex spaces depending on the threshold for determining how many consecutive columns are regarded as a separator of space. If we set the threshold at four columns, we would have two convex spaces; if we set the threshold at two columns, we would have three convex spaces. Such threshold is, again, mostly arbitrary.

[2] Convex partitioning is equivalent to finding minimum clique cover of visibility graph of the floor, one of the well-known graph theoretical problem because a convex space is equivalent to maximal clique in visibility graph. Minimum clique cover is known to be NP-complete.

[3] If the columns are circular ones, the columns should be approximated to straight lines before we apply convexity rule. We get 12 convex spaces when the columns are approximated to rectangular shape. If the columns are approximated to octagon, much more convex partitions would be required.

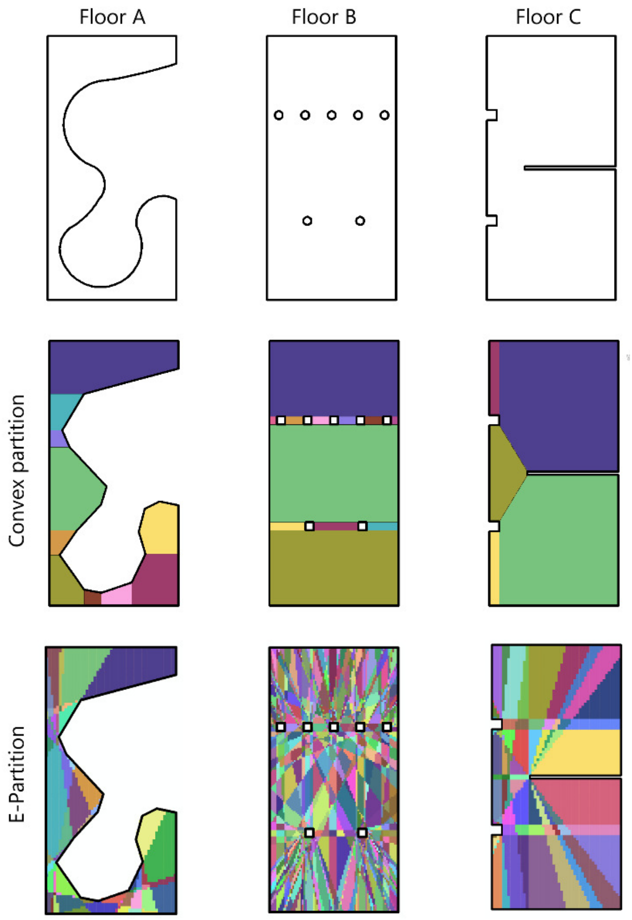


Figure 2: Exemplar floors for traditional methods

We may think we can address the problem of arbitrariness by listing 'exception rules' related to convexity condition such as to ignore small indents less than one meter's offset; to straighten concave walls with one meter's segments; to ignore the thickness of a wall less than 0.3 meter; to approximate a circular column as a rectangular column; to ignore the spaces between columns closer than one meter and so on. However, to list complete set of exception rules seems not an easy task, and the rules may conflict each other. Moreover, to translate such rules into machine-understandable language looks even difficult. Partly because of the lack of well-defined set of machine-understandable exception rules, and partly because of inherent computational complexity of convex partitioning, there is no widely used computer program for the automation of convex decomposition to our best knowledge. Thus convex partitioning still conducted manually and is remained vulnerable to operational arbitrariness.

E-Partition

E-partition is a method of dividing spaces into e-spaces that are 'informationally stable' in the sense that any location in the e-space shares the same set of visible vantage points such as corners or end points of walls (Peponis et al. 1997). This has been the most rigorous approach seeking the minimum 'unit' of space based on visual information. The method provides a mathematically well-defined set of subspaces, and its computational complexity is significantly lower than that of convex partitioning.

In spite of its theoretical elegance, e-partition method has some limitations in its application.

Firstly, the method is difficult to apply to a floor with curved walls or boundaries that have no salient vantage points. In such a case, we need a rule for designating a point on the curve as a vantage point or a rule for converting the curve into straight line segments as we did for Floor A in Figure 2.

Secondly, in many practical cases, e-partition method generates a large number of tiny e-spaces usually unsuitable for common analytic purposes. This is because the number of e-spaces dramatically increases and hence the size of e-space rapidly shrinks as the complexity of space increases[4]. Although having many small partitions help a researcher to conduct a fine-grained analysis, too many partitions make the floor's spatial structure almost illegible (see e-partitions of Floor B in Figure 2).

3. Proposed Method

The core idea of this method is to decompose the visibility graph of a floorplan into closely interconnected groups of nodes. Then, the key becomes how to find such groups from the network and how good would be the grouping. For this aim, this study utilizes a community detection technique developed for discovering closely related groups in a network. Later on, the new method proposed in this study will be called NCVG (Network Communities in the Visibility Graph) method.

3.1 Visibility graph

Unlike convex partitioning or e-partitioning that 'cuts' a floorplate into polygons with dividing lines to make subspaces, this method defines a subspace with a set of grid points on the floorplate in the same way that an isovist from a point can be defined as a set of points that are visible from the point. This approach that defines an area with a point set is also applicable to traditional methods like convex partitioning or e-partitioning. For example, a convex space can be defined as a set of grid points that forms a clique, or a complete graph. An e-space can be defined as a set of grid points sharing the same set of visible vantage points. As seen in the point-set based representations of a convex space and an e-space, the key process to identify a subspace is to identify a group of grid points that are closely related or that share the similar attributes. This method finds subspaces by identifying closely related groups of grid points using network structure of the visibility graph of a floor.

3.2 Community detection

Figure 3 shows a network structure with three obvious groups of nodes in the network. Each group has relatively many edges inside the group and fewer edges going outside of the group. Such a group is often called a 'community' by network scientist, which is usually defined as "a cohesive group of nodes that are connected 'more densely' to each other than to the nodes in other communities" (Porter, Onnela, and Mucha 2009). Finding a community structure of a network has been proved to be useful in many fields. In biology, for example, community detection algorithms have been applied to protein-protein interaction networks to identify functional modules of proteins (Chen and Yuan 2006). In communication network, by analyzing community structure of an email exchange network in scientific labs, researchers could identify groups of people quite closely matched to the labs' organizational structure and project assignment (Tyler, Wilkinson, and Huberman 2005). An analysis on the community structure of

[4] In general, the number of e-spaces is proportional to the quadruple of the number of vantage points. The number of 'cutting edges' is proportional to the square of the number of vantage points, and the number of sliced plane (e-space) is proportional to the square of the number of cutting edges.

a citation network over 600 scientific journals presented a 'map of science' showing how each discipline in science is related to others (Rosvall and Bergstrom 2008).

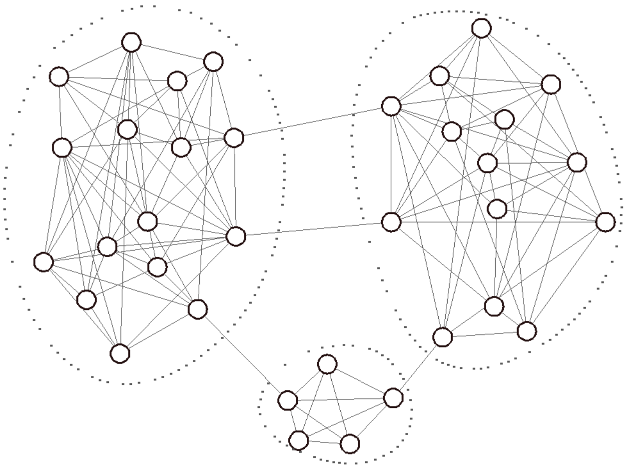


Figure 3: A network with obvious community structure

There is also a traditional way of decomposing a network into subcomponents, which is often called graph partitioning. Unlike the community detection methods, graph partitioning requires to fix the number of clusters to be separated in advance (Newman 2006). The main goal of graph partitioning is to find the best division of network with the given number of divisions; hence, this approach is useful when we have a strong reason for the pre-fixed number of divisions. In this paper, we will not follow this line of approach as we do not have such a reason for the number of clusters in general spatial decomposition. So we will place more focus on community detection techniques, which does not require the number of clusters in advance.

Modularity

The crux of community detection is how to find 'good' partitions of a network. One approach is to set up a quality function measuring how good current partitioning is and then to optimize the quality function. Currently, one of the most widely used such a quality function is Newman's *modularity* Q (Newman and Girvan 2004):

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j)$$

where m is the total number of edges in the network, A is the adjacency matrix of the network, k_i is degree of vertex i, $\delta(c_i,c_j)$ is the delta function whose value is 1 when vertex i and j belong to the same cluster and 0 otherwise. In plain words, the modularity function Q measures the difference between the existing number of edges in the cluster and the expected number of edges in the cluster when the network is randomly wired ignoring community structure (Newman 2006). Thus, maximizing modularity Q means minimizing the difference between the number of intra-group edges and the expected number of inter-group edges (Fortunato 2010).

A critical difference between modularity based decomposition and convex partitioning is that convex partitioning does not consider how inter-group edges are distributed across groups. Look at Figure 4 showing a floor with three rooms connected with two openings; one is wide and another is narrow. Without any complex analysis, we can say that room A and room B form almost one space, whereas room B and room C are quite separated. Convex partitioning ignores such an obvious difference between the two openings and yields three convex spaces (if wall thickness is ignored).

The difference between the two openings becomes clearer when we construct a visibility graph on the floorplan in Figure 4. There are much more inter-edges passing the opening between A and B (total 311 edges) than inter-edges passing the opening between B and C (total 181 edges). Thus, modularity based partitioning prioritizes B-C cut to A-B cut, whereas no priority is possible to be given by convex partitioning.

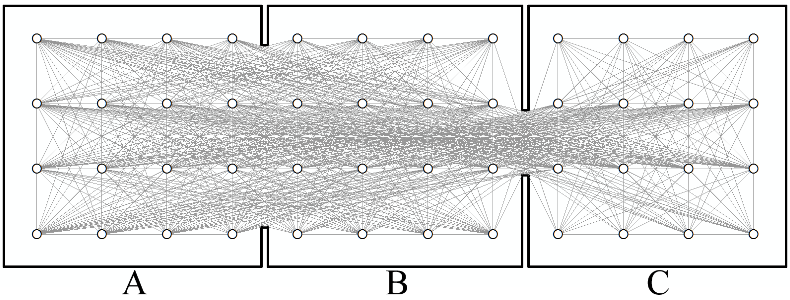


Figure 4: Floor with two thresholds

Another noteworthy difference in modularity based decomposition is that it gives a decomposition based on the global property of a spatial network, while convex partition or e-partition uses only the local property of the space. For example, whether a space is convex or not has nothing to do with other parts of a floor. It can be determined only with local information. In contrast, modularity-based decomposition cannot be done without the knowledge of the whole network structure. For example, let's assume that the rooms in Figure 4 are only a part of a large floor. Then, whether there would be any partition inside A-B-C cannot be determined without information on other spaces connected to A-B-C. If there are plenty of good locations for 'cuts', then we would be less likely to have a partition within A-B-C. On the contrary, if outside spaces are all very well-connected, then we would be more likely to have a partition within A-B-C.

VOS clustering technique

Among many community detection algorithms proposed so far by network scientists, this paper uses VOS clustering technique (Van Eck and Waltman 2007) for finding communities in space. This is because the method can adjust 'resolution' of analysis, respects edge weights, and has reasonable computation cost. VOS technique has a slightly different form of quality function from Newman's modularity Q in order to introduce a resolution parameter γ and the degree of association strength between nodes (edge weight). The quality function V of VOS technique is:

$$V = \frac{1}{2m}\sum_{ij}[s_{ij} - \gamma]\,\delta(c_i, c_j)$$

where s_{ij} denotes association strength between vertex *i* and *j*, which is given by

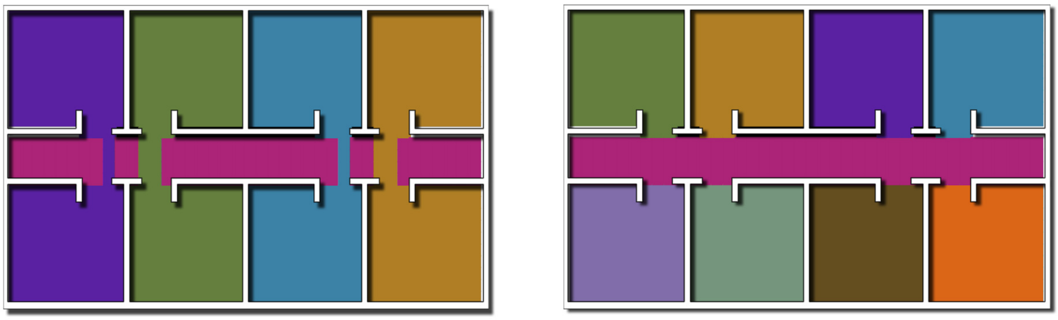
$$s_{ij} = \frac{2mk_{ij}}{k_i k_j}$$

where k_{ij} denotes the number of edges between vertex i and j.

It has been shown that the quality function V of VOS clustering technique is equivalent to Newman's modularity Q when resolution parameter γ and edge weights are set to one (Waltman, van Eck, and Noyons 2010).

3.3 Distance weight

Panel (a) in Figure 5 shows a sample result of VOS partitioning (γ=0.4) applied to a typical floor with a corridor. We can see the upper rooms and the lower rooms are pairwisely interconnected. This is, although not desirable, but understandable result with the given visibility graph structure that connects any visible grid points *equally* no matter how near or far they are. In other words, the visibility graph does not contain information about the 'locality'. Hence, a distant pair of points across the corridor can be grouped together because they are visually 'neighbors' equally as a closely located pair of points within the same room. This is not what we expected to get from floorplan decomposition.



(a) No distance weight ***(b) Inversely distance weighted***

Figure 5: Comparison between no weight visibility graph and distance weighted visibility graph

The problem can be solved by emphasizing the locality of a visibility graph. In other words, one way of to strengthen locality is to make ties between nearer points stronger and to make the ties between farther points weaker. Thus, we give edge weights to the visibility graph according to the inverse of the distance between the two points. Panel (b) in Figure 6 is the result from the same algorithm with panel (a) except for the edge weights. It gives a set of partitions fit with our intuitive partitioning.

3.4 Implementation

We developed software for calculating and visualizing NCVG method. ArcPy, the official Python package for ArcGIS by ESRI was used for generating grid points, checking visibility, and visualizing the decomposition result. NetworkX, an open-source Python package for network analysis was used for general network operation such as constructing a visibility graph, giving edge weights, and exporting a graph. VOS clustering software by van Eck and Waltman was used for the calculation of VOS algorithm. Python was used for a 'glue' language combining these components.

The software gets shapefiles of a floor's boundary and internal walls as inputs. Then it builds a weighted visibility graph and determines groups of nodes in the visibility graph according to the given resolution γ. The outputs are a shapefile containing grid points with partition information and an image file visualizing the partitions.

4. Discussion

4.1 Revisiting the three floors

Figure 6 shows a result obtained by applying NCVG method (γ =0.4) to the three floorplans previously shown in Figure 2. Floor A with a curved wall is decomposed into four subspaces being separated at "bottleneck" locations. Floor B with seven free standing columns is successfully separated into three subspaces. In floor C, the algorithm ignores small indents in each room and identifies two rooms following the designer's intent.

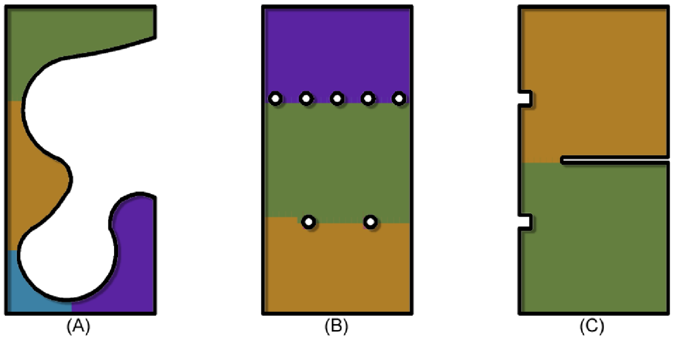


Figure 6: Application of NCVG to the three previous floors

One might ask, "Can we have more (or less) divisions for Floor A?" Or, "The upper part of Floor B is separated by five columns, while the lower is separated only by two columns. Should we have to treat them equally?" Or, "What would happen if the indents in Floor C become larger? Do we still have two subspaces?" These questions are closely related to the effect of resolution parameter.

4.2 Effect of resolution

One of the major advantages of NCVG method is that we can control the resolution of analysis. For example, if we need to conduct a fine-grained analysis of a floor, we would use a higher resolution parameter that identifies more groups. Let's look at an exemplar floorplan. How many subspaces would you expect to see from the floor (a) in Figure 7?

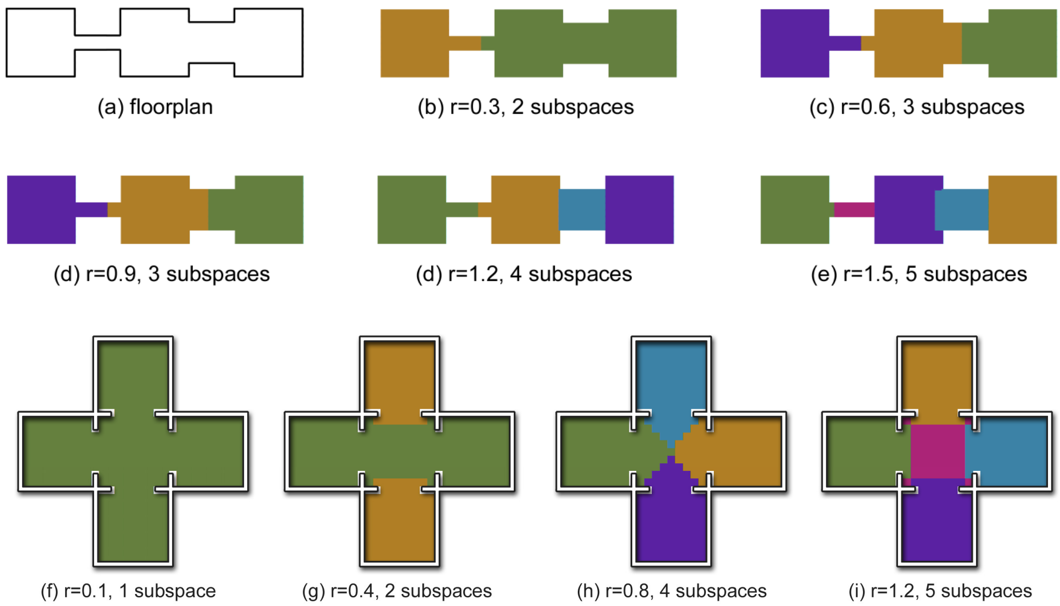


Figure 7: Increased number of subspaces as resolution increases

The panels from (a) to (e) in Figure 7 show how subspaces are differentiated by the level of resolution. If a researcher wants to see a big picture or to identify the most critical 'cuts', he or she would set the resolution low enough and get two subspaces as in Figure 7-(b), saying, "What is important in this spatial structure is the long bottleneck in the left part, and we should ignore the trivial indents in the right part." Or, he or she would choose panel (e) if he or she thinks, "There are three rooms and two connectors in this space. They are all independent spaces and I'd like to see them all."

Going back to Figure 6, we can predict how the three floors will be decomposed as resolution changes. For floor A, we will have more divisions as the resolution parameter increases. Floor B will have only one subspace with very low resolution. As the resolution parameter increases, the upper part of Floor B will be identified firstly and then the lower part will be also identified. The small indents in Floor C will be ignored when the resolution parameter is small. However, when resolution increases or the size of indents becomes larger, the floor will be divided into more than two subspaces because the indents are not to be ignored.

4.3 Hierarchical spatial structure

One useful aspect of the series of decomposition with changing resolution is to reveal underlying spatial structure. Figure 8 shows the series of spatial differentiations from a simplified Miesian house. With a low resolution parameter (γ =0.10), the floor is divided into the three subspaces: MNOP – QR – STU. As the resolution γ increases to 0.30, the largest part MNOP is differentiated into three subspaces M-OP-N. At γ = 0.40, STU cluster is divided into S and TU. At γ = 0.80, OP is divided into two subspaces, O and P. At γ =1.20, Q becomes independent from R. Finally, T and U is differentiated at γ =1.40.

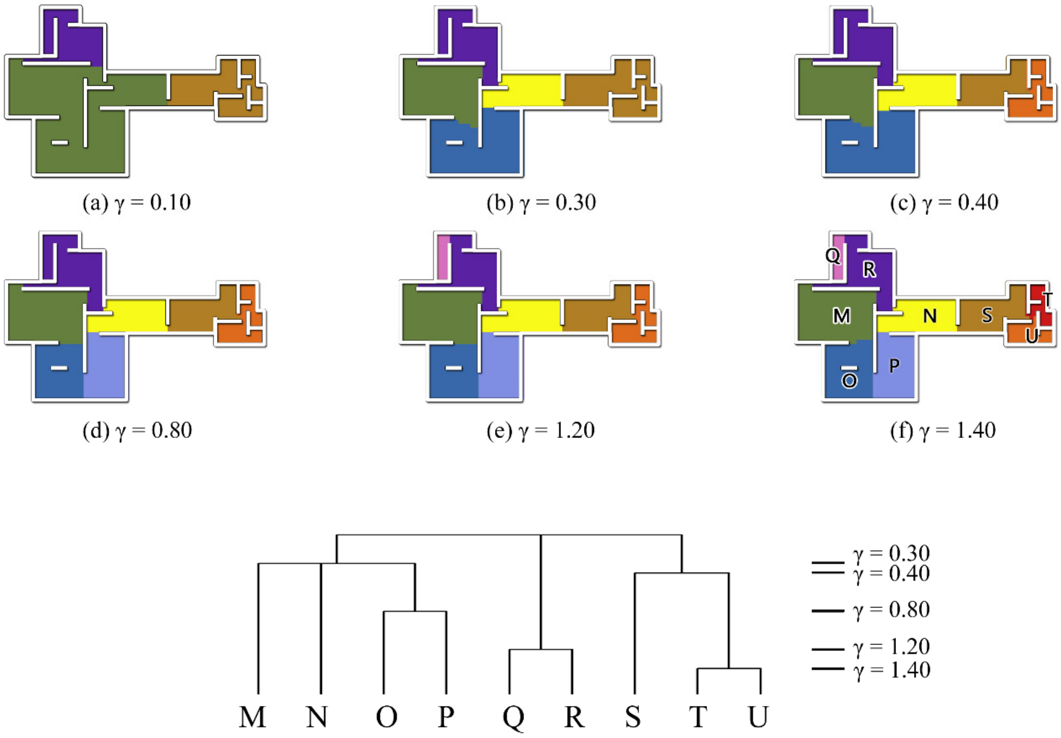


Figure 8: Differentiation of Space

The differentiation process is summarized as a dendrogram in Figure 8. The dendrogram gives a picture on the hierarchical spatial structure of the floor, which is not easily captured in the traditional graph representation of floorplan. Space S is adjacent both to N and TU; however, S-N connection and S-TU connection are not equal. S is more closely related to TU than N in that S-TU is merged together earlier than S-N. Likewise, space P that is adjacent both to N and O is merged first with O and then merged with N, which means P-O connection is stronger than P-N connection.

5. Conclusion: limitations and further work

In this paper, we present a new method for the decomposition of space requiring no ad-hoc rules and relatively free from an operator's arbitrariness. Unlike traditional methods, this method decomposes space based on the global property of a spatial network by adopting modularity function as a quality function of decomposition. Also, VOS technique used for this method enables researchers to adjust the level of analytic resolution. This gives much more flexibility in the analysis of space. Also, this method provides an analytic tool for exploring spatial hierarchy by the examination of the series of spatial differentiations.

Although this paper provides an interesting way of spatial decomposition, much work is still required. First of all, it will be interesting if the result of the decomposition using this method is compared to human recognition of space or actual space utilization. Secondly, a dedicated algorithm for community detection in spatial network may need to be developed. Currently we are using VOS algorithm developed originally for citation network, not for spatial network. We may expect better performance and more control on the community detection process if we have a tailored algorithm for spatial network. Thirdly, applying this method to three dimensional space might be interesting. One of the strength of this method is that this method can be expanded to three dimensional environment with almost no additional effort once three dimensional visibility graph is given.

Finally, we may explore the possibility of applying community detection techniques to spatial networks other than visibility graphs. For example, we may apply community detection techniques to the road network of a city in order to identify 'natural' urban tissues of the city and to explore how such urban tissues are well or poorly matched to actual urban behaviors. Also, we may apply the techniques to the convex map of a large building complex to identify well-clustered parts of the building complex.

References

Chen, Jingchun, and Bo Yuan. 2006. "Detecting Functional Modules in the Yeast Protein–protein Interaction Network." *Bioinformatics* 22 (18): 2283–2290.

Fortunato, Santo. 2010. "Community Detection in Graphs." *Physics Reports* 486 (3-5): 75–174.

Hillier, Bill, and Julienne Hanson. 1984. *The Social Logic of Space*. Cambridge University Press.

Karp, Richard M. 1972. "Reducibility among Combinatorial Problems." In *Complexity of Computer Computations*, edited by R. E. Miller, J. W. Thatcher, and J. D. Bohlinger: 85–103. The IBM Research Symposia Series. Springer US.

Newman, M. E. J. 2006. "Modularity and Community Structure in Networks." *Proceedings of the National Academy of Sciences* 103 (23): 8577–8582.

Newman, M. E. J., and M. Girvan. 2004. "Finding and Evaluating Community Structure in Networks." *Physical Review E* 69 (2): 026113.

Peponis, J., and J. Wineman. 2002. "Spatial Structure of Environment and Behavior." *In Handbook of Environmental Psychology*: 271–291.

Peponis, J., J. Wineman, M. Rashid, S. Kim, and S. Bafna. 1997. "On the Description of Shape and Spatial Configuration Inside Buildings: Convex Partitions and Their Local Properties." *Environment and Planning B* 24: 761–782.

Porter, Mason A., Jukka-Pekka Onnela, and Peter J. Mucha. 2009. "Communities in Networks." *Notices of the AMS* 56 (9): 1082–1097.

Rosvall, Martin, and Carl T. Bergstrom. 2008. "Maps of Random Walks on Complex Networks Reveal Community Structure." *Proceedings of the National Academy of Sciences* 105 (4): 1118–1123.

Turner, A., and Alan Penn. 1999. "Making Isovists Syntactic: Isovist Integration Analysis." *In 2nd International Symposium on Space Syntax*, Brasilia.

Tyler, Joshua R., Dennis M. Wilkinson, and Bernardo A. Huberman. 2005. "E-Mail as Spectroscopy: Automated Discovery of Community Structure within Organizations." *The Information Society* 21 (2): 143–153.

Van Eck, Nees Jan, and Ludo Waltman. 2007. "VOS: A New Method for Visualizing Similarities Between Objects." *In Advances in Data Analysis*, edited by R. Decker and H.-J. Lenz: 299–306. Studies in Classification, Data Analysis, and Knowledge Organization. Springer Berlin Heidelberg.

Waltman, Ludo, Nees Jan van Eck, and Ed C.M. Noyons. 2010. "A Unified Approach to Mapping and Clustering of Bibliometric Networks." *Journal of Informetrics* 4 (4): 629–635.